## Remarks

Applicant wishes to thank the Examiner for withdrawing the previous § 112 rejections.

I. Claim Rejections under 35 U.S.C. § 103.

Claims 1-3, 8-16, and 21-30 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication US 2003/0005419 (Pieper) in view of *Portable Software Library Optimization*, 2/1998 (Cain).

**A. Limitations regarding "based on results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language".**

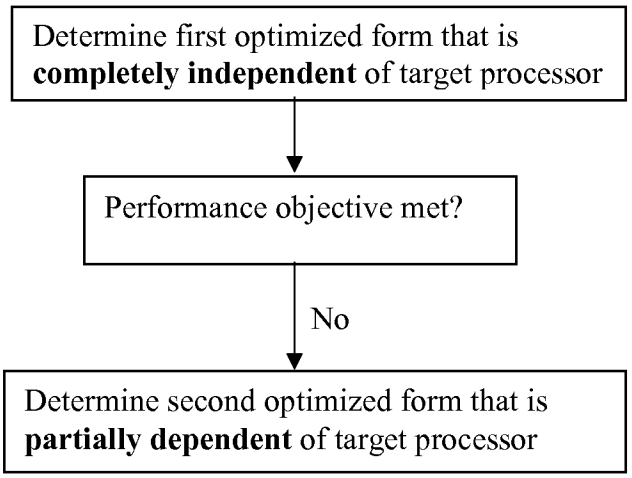
Claim 1 recites:

> (a) optimizing the software program such that *a resulting first optimized form of the software program is completely independent* of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;
>
> (b) based on results of comparing the first performance profile with the performance objectives, *if the performance objectives are not met by the first optimized form of the software program*, then optimizing the first optimized form of the software program such that *a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor* and is coded in the high-level language , wherein the at least one portion of the second optimized form of the software program is less than an entirety of the second optimized form
>
> (Emphasis Added)

Claims 14 and 31 recite similar limitations. Pieper and Cain do not disclose or suggest the above limitations.

As an initial matter, it is important to note that claim 1 requires (1) determining a first optimized form of the software program that is **completely independent** of target processor, and

then (2) determining a second optimized form that is **partially dependent** (having at least one portion less than an entirety that is dependent) on the target processor only *if the performance objectives are not met by the first optimized form*, as follow:




The above features allow the optimized form of the software program to be progressively more dependent on the target processor based on whether performance objective is met or not (i.e., going from completely independent to partially dependent if performance objective is not initially met by the first optimized form that is less target dependent than the second optimized form), as required in claim 1.

Pieper does not disclose or suggest the above features. Rather, Pieper discloses performing an optimization process 58 to generate code 60 that is "substantially independent of the architecture of the target processor 12" (see paragraph 30, and figure 2 reproduced below with markup). On pages 9-10 of the Office Action, the Examiner has interpreted the code 60 that is "substantially independent" of target processor in Pieper as having a portion that is dependent (partially dependent) on the target processor. However, even based on this characterization, Pieper still does not disclose or suggest the above features. This is because in Pieper, the optimization process 58 to generate code 60 follows step 56 (see figure 2). Thus, in Pieper, the code 60 is always generated, and it is not generated *based upon whether performance objective is met by a first optimized form that is completely independent of target processor*, as required in claim 1.

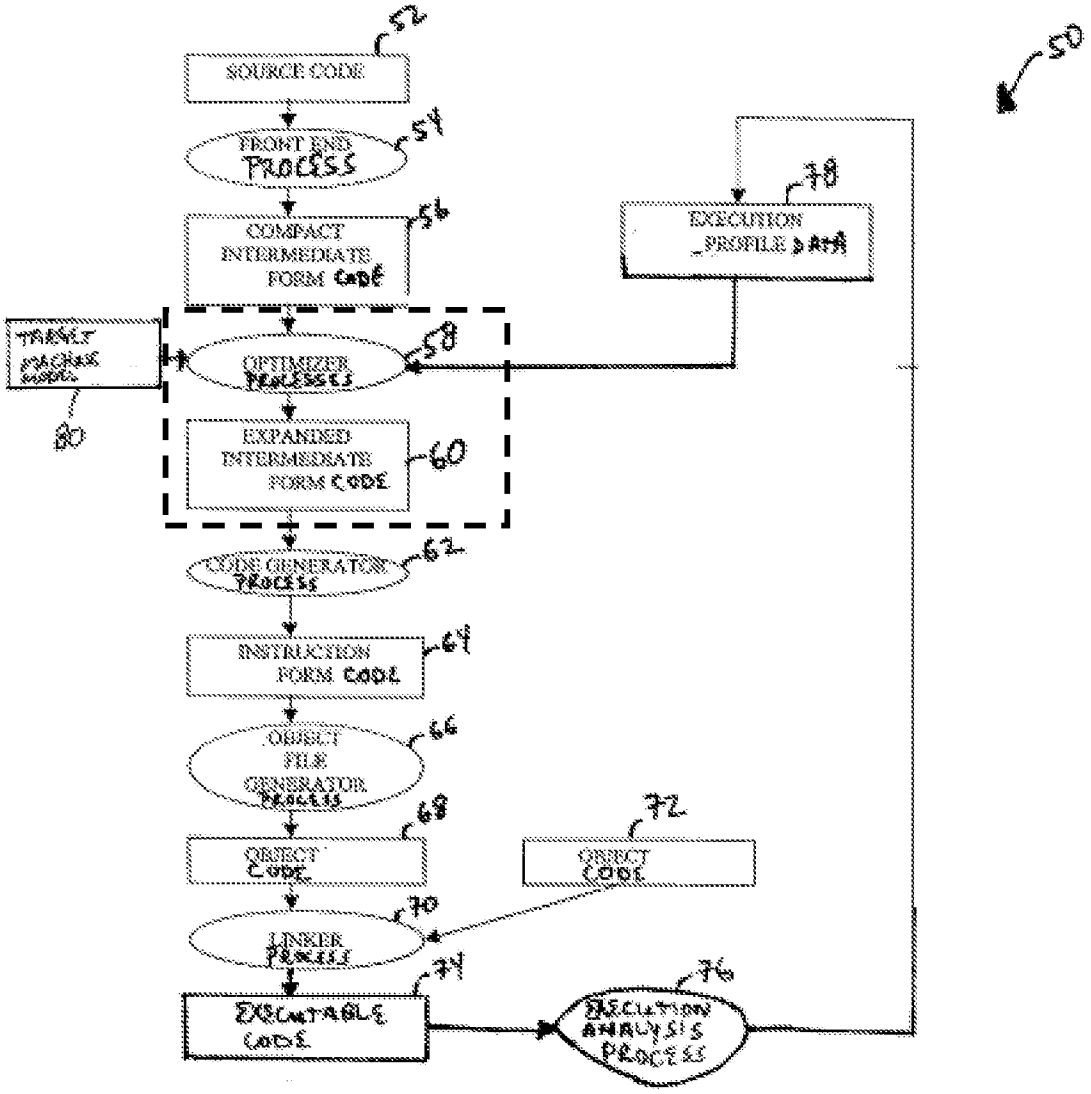



FIG. 2

Also, Pieper discloses that the optimization process 58 is repeated based on execution profile data (see figure 2 above, showing optimization process 58 has input from element 78). Notably, the execution profile data in element 78 is generated based on code 64 that is completely *dependent* on target processor (see paragraph 31, describing that code 60 is translated to instruction 64 so that it is dependent on target processor, and figure 2, showing that element 78 flows from element 64). Thus, in Pieper, the repeating of optimization process 58 to generate code 60 is based on execution profile

data derived from code 64 that is *dependent* on target processor, and is not based on whether objective profile is met by a first optimized form that is "completely independent" on target processor, as required in claim 1. In fact, since Pieper specifically teaches that code 60 is based on profile data derived from target processor *dependent* code 64, Pieper actually teaches away from generating code 60 (which the Examiner analogized as the claimed "a second optimized form") based on performance profile from a code that is "completely independent" on target processor (note that the claims require that the second optimized form be conditioned upon whether objective performance is met by a first optimized form that is "completely independent" on target processor). Note that no prima facie case of any § 103 rejection can be maintained if a cited reference teaches away from a claimed feature.

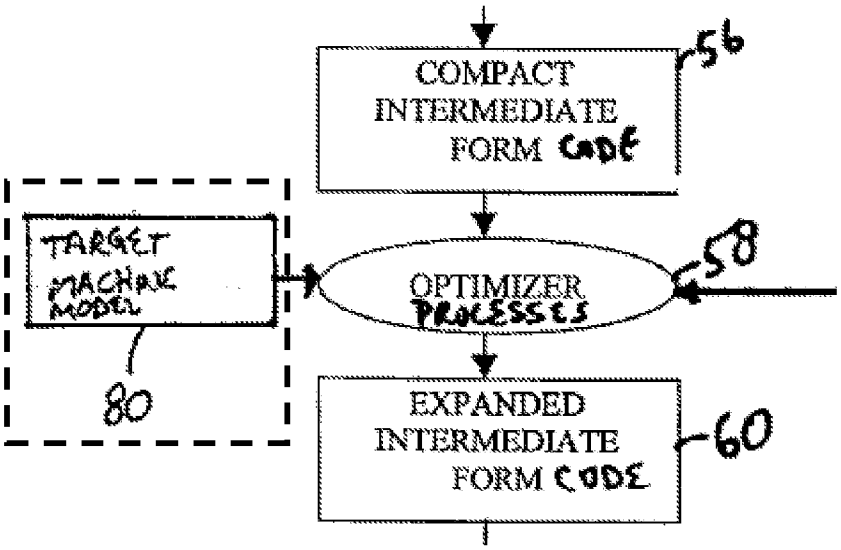
Also, Applicant respectfully notes that figure 2 and its corresponding passage of Pieper disclose repeating the optimization process 58 to make the code 60 more "efficient", and therefore, the repeat of optimization process 58 is not to make the code 60 more dependent on target processor). This is further evidenced from the fact that in each cycle of optimization, the code 60 (which the Examiner considers to be partially dependent on target processor) is always "substantially independent", and the code 60 does not become more target dependent after each repeat of the optimization process 58. Thus, to the extent that the execution profile data 78 in Pieper is analogized as the claimed "performance objectives" or "performance profile", Pieper discloses that the code 60 is made more "efficient", not more target dependent, based on the profile data 78.

Applicant further notes that Pieper does disclose translating code 60 that is "substantially independent" to code 64 that is target "dependent". However, in Pieper, the code 60 is always translated to code 64, regardless of whether optimization objective is met or not (note that there is no performance objective evaluation step between elements 60 and 64 indicating that code 64 is generated only if performance objective is not met by code 60). Thus, to the extent that code 60 and code 64 are analogized as the claimed "first optimized form" and "second optimized form", respectively, Pieper clearly does not disclose or suggest that the second optimized form (code 64) is conditioned whether performance objective is met by the first optimized form 60 (that is less target dependent than the second optimized form).

Cain also does not disclose or suggest the above limitations, and therefore fails to make up the deficiencies present in Pieper. Since both Pieper and Cain do not disclose or suggest the above limitations, any purported combination of these references cannot result in the subject matter of

claims 1, 14, and 31. For at least the foregoing reasons, claims 1, 14, and 31, and any claims depending therefrom, are believed allowable over the cited references of record.

Furthermore, Applicant certainly agrees with the Examiner on page 3 of the Office Action that Pieper does not disclose that a first optimized form of a software program is "completely independent" of a target processor. However, Applicant respectfully disagrees that it would have been obvious to modify Pieper's technique to incorporate a first optimized form of a software program that is completely independent (as allegedly disclosed in Cain), as purported in the Office Action. This is because Pieper specifically requires that a "target machine model 80" be input into the optimization process 58 (see partial figure 2 reproduced below with markup to show the machine dependent aspect).




Thus, the process of Pieper cannot possibly incorporate any optimized form of a software program that is "completely independent" of a target processor. For the foregoing reasons, the purported modification of Pieper would render the method of Pieper inoperable for its intended purpose. Note that a prima facie case of any § 103 rejection cannot be established if a purported modification of a reference would render the reference inoperable for its intended purpose. For these additional reasons, claims 1, 14, and 31, and any claims depending therefrom, should be allowable over the cited references of record.

**B. Limitations regarding "flagging".**

Claim 1 also recites *flagging* the at least one portion to indicate that the at least one portion is dependent on the target processor *if the first optimized form of the software program is optimized to create the second optimized form of the software program* (Emphasis Added). Claims 14 and 31 recite similar limitations. Applicant agrees with the Examiner that Pieper does not disclose the above limitations. Applicant respectfully notes that there is nothing in Cain that discloses or suggests that any act of flagging is *conditioned upon* whether "the first optimized form of the software program is optimized to create the second optimized form of the software program" as described in the claim (i.e., note the limitation "if" in the claims).

According to page 10 of the Office Action, Cain allegedly discloses "#if-define", which is also described in the subject application. However, Applicant respectfully notes that there is nothing in Cain that discloses or suggests that the "#if-define" is a flag that is conditioned upon whether "the first optimized form of the software program is optimized to create the second optimized form of the software program," as described in the claims. The above argument does not appear to have been addressed in the current Office Action.

In addition, Applicant respectfully submits that a mere disclosure of "#if-define" in Cain does not amount to any disclosure that a flagging is conditioned upon whether "the first optimized form of the software program is optimized to create the second optimized form of the software program," as described in the claims.

For these additional reasons, claims 1, 14, and 31, and any claims depending therefrom, are believed allowable over the cited references of record.

## CONCLUSION

If the Examiner has any questions or comments regarding this response, please contact the undersigned at the number listed below.

To the extent that any arguments and disclaimers were presented to distinguish prior art, or for other reasons substantially related to patentability, during the prosecution of any and all parent and related application(s)/patent(s), Applicant(s) hereby explicitly retracts and rescinds any and all such arguments and disclaimers, and respectfully requests that the Examiner re-visit the prior art that such arguments and disclaimers were made to avoid.

The Commissioner is authorized to charge any fees due in connection with the filing of this document to Vista IP Law Group's Deposit Account No. **50-1105,** referencing billing number **00PA339US03**. The Commissioner is authorized to credit any overpayment or to charge any underpayment to Vista IP Law Group's Deposit Account No. **50-1105,** referencing billing number **00PA339US03**.

Respectfully submitted,

DATE: July 26, 2011

By: /Gerald Chan/
Gerald Chan
Registration No. 51,541

VISTA IP LAW GROUP, LLP
1885 Lundy Ave., Suite 108
San Jose, California 95131
Telephone: (408) 321-8663 (Ext. 203)
Facsimile: (408) 877-1662